REMARKS

Claims 7, 9, 11, 15, 18-20, 22, and 26-41 are pending. Claim 27 stands rejected under 35 U.S.C. § 112, ¶ 1 as failing to comply with the written description requirement. Claims 7, 9, 11, 26-28, and 36 stand rejected under 35 U.S.C. § 112, ¶ 2 as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. Claims 7, 9, 11, 15, 26-28, and 32-41 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,125,391 to Meltzer et al. in view of U.S. Patent No. 6,772,396 to Cronin et al. Claims 18-20, 22, and 29-31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,125,391 to Meltzer et al. in view of U.S. Patent No. 6,772,396 to Cronin et al. and U.S. Patent No. 6,622,170 to Harrison et al.

Reconsideration is requested. No new matter is added. The specification is amended. Claims 7, 15, 26, 32, 36, and 39-40 are amended. Claim 28 is canceled. The rejections are traversed. Claims 7, 9, 11, 15, 18-20, 22, 26-27, and 29-41 remain in the case for consideration. The Examiner is requested to treat any changes to the claims not reflected as strikethrough or underline text as inadvertent typographical errors.

The Applicant points out that the U.S. Patent & Trademark Office continues to mail papers for this application to the wrong address. Although the last paper was addressed to Haynes and Boone, LLP, the Haynes and Boone firm no longer represents the Applicant in this matter. The Applicant has attached a copy of the Power of Attorney showing that the firm of Marger Johnson & McCollom, P.C. now represents the Applicant, with a new mailing address. While the Haynes and Boone firm has graciously continued to forward papers to the Marger Johnson firm in this matter, **the undersigned respectfully requests that the U.S. Patent & Trademark Office correct the mailing address to the Marger Johnson firm, as indicated in the attached copy of the Power of Attorney. The Examiner is also requested to contact the undersigned to schedule an interview for this patent application.**

REJECTION OF CLAIMS UNDER 35 U.S.C. § 112, ¶ 1

The Examiner rejected claim 27 as failing to satisfy the written description requirement because claim 27 recites "the first predetermined format and the second predetermined format are the same predetermined format". While the Applicant acknowledges that this exact wording is not in the specification, the Applicant believes there is support for this claim.

The concept captured by claim 27 is the idea that multiple applications might share a common predetermined format. The Examiner is referred to page 10, lines 20-22 of the specification. At that place, the specification recites: "This application data format may be specific to the particular application 102, or may be another XML format." The first predicate of the disjunction ("or") in this sentence covers situations where the predetermined format is specific to a particular application; by implication, the other predicate of the disjunction applies when the data format is not application-specific. In other words, two applications either have different or common predetermined formats: the latter is exactly what claim 27 recites.

It is also worth noting that there are many well-known examples in the art of applications that share data formats: for example, the applications that comprise the Microsoft Office suite. First, it is quite common (although not absolutely required) for different versions of an application to use the same data format. In addition, applications that are bundled together in a suite often use a common interface: clearly, such applications also use common data formats.

REJECTION OF CLAIMS UNDER 35 U.S.C. § 112, ¶ 2

Claims 7, 26, and 36 have been amended to correct typographical errors, which should address the rejections under 35 U.S.C. § 112, ¶ 2.

REJECTION OF CLAIMS UNDER 35 U.S.C. § 103(a)

Referring to claim 7, the invention is directed toward a software program for facilitating the use of a distributed directory running in a computer network, the program comprising being stored on a recordable medium and including instructions for: receiving an event from the distributed directory into an XML generator, the distributed directory including at least one resource on one of at least two computers on the computer network; converting the event into XML data representing the event; transforming the XML data representing the event to a first predetermined format by a transformation processor using a first stylesheet, the first predetermined format being responsive to a first application running in the computer network; transmitting the transformed XML data representing the event to the first application; transforming the XML data representing the event to a second predetermined format by the transformation processor using a second stylesheet, the second predetermined format being responsive to a second application running in the computer network; and transmitting the transformed XML data representing the event to the second application.

Referring to claim 15, the invention is directed toward a software program for facilitating the use of a distributed directory running in a computer network, the program comprising instructions for: receiving a first event from a first application in a first native application format; converting the first event into markup language data; transforming the first event to a predetermined format by a transformation processor using a transformation profile, the predetermined format being responsive to the distributed directory, the transformation profile including formatting instructions for transforming the markup language data to the predetermined format, the distributed directory including at least one resource on one of at least two computers on the computer network; transmitting the transformed first event to the distributed directory; receiving a second event from a second application in a second native application format; converting the second event into markup language data; transforming the second event to the predetermined format by the transformation processor using the transformation profile; and transmitting the transformed second event to the distributed directory.

Referring to claim 18, the invention is directed toward a distributed computer system comprising: a first processor connected to a network for executing computer code; a second processor connected to the network for executing computer code; a first memory connected to the first processor; a second memory connected to the second processor; a distributed directory, wherein first and second portions of the distributed directory are stored in the first memory and the second memory, respectively; a first application, a portion of which being stored in one of the first memory and the second memory; a second application, a portion of which being stored in one of the first memory and the second memory; a first transformation profile defining a first predetermined format for use by the first application; a second transformation profile defining a second predetermined format for use by the second application; software for detecting a directory event in the distributed directory; software for transforming the directory event to the first predetermined format by using a generic transformation tool and the first transformation profile; software for transforming the directory event to the second predetermined format by using the generic transformation tool and the second transformation profile; software for providing to the first application the directory event transformed to the first predetermined format; and software for providing to the second application the directory event transformed to the second predetermined format.

Referring to claim 32, the invention is directed toward a method for interfacing with a distributed directory in a computing system, comprising: providing a first transformation profile defining a first predetermined format for use by a first application; providing a second transformation profile defining a second predetermined format for use by a second application; detecting an event in the distributed directory, the distributed directory including at least one resource on one of at least two computers on a network; transforming the event to the first predetermined format by using a transformation tool and the first transformation profile; transforming the event to the second predetermined format by using the transformation tool and the second transformation profile; providing to the first application the event transformed to the first predetermined format; and providing to the second application the event transformed to the second predetermined format.

Referring to claim 40, the invention is driver infrastructure for interfacing a distributed directory and applications comprising: a generator to receive a directory event from the distributed directory and to generate a generic data for the directory event, the distributed directory including at least one resource on one of at least two computers on a network; a first transformation profile defining a first predetermined format for use by a first application; a second transformation profile defining a second predetermined format for use by a second application; a transformation processor to transform the generic data for the directory event into a first application data for the first application using the first transformation profile and to transform the generic data for the directory event into a second application data for the second application using the second transformation profile; and a transmitter to transmit the first application data to the first application and to transmit the second application data to the second application.

In contrast, Meltzer teaches market makers using documents in trading partner network. As shown in FIG. 15, Meltzer receives documents via a communications agent. The documents can be in any syntax. Meltzer converts the documents to XML, then uses an XML parser to verify that the documents are properly formatted in XML. Using the business interface definition compiler (BIDC), the documents are compiled into Java documents. The Java documents are then delivered to the enterprise solutions using the document service.

The Examiner relies on Cronin solely for the concept of stylesheets and transformation profiles, and on Harrison solely for the concept of portions of the distributed directory being stored in multiple memories. The Applicant believes that neither Cronin nor Harrison teach or suggest any other features of the claimed invention.

In rejecting the claims, the Examiner has asserted that "Meltzer teaches receiving an event from a server into an XML generator . . ." (Office Action dated August 24, 2004, page 4, ¶ 9). The Applicant does not disagree with the Examiner about the teaching of Meltzer, but the Applicant believes that the Examiner's argument that Meltzer teaches the claimed invention is incorrect. Meltzer teaches events coming from a communication agent (*see* FIG. 15 of Meltzer).

In making the rejection, the Examiner is clearly analogizing applications to directories. This analogy is not justified. An application is "[a] complete, self-contained program that performs a specific function directly for the user. This is in contrast to system software such as the operating system kernel, server processes and libraries, which exists to support application programs. . . . One distinction between an application program and the operating system is that applications always run in 'user mode' (or 'non-privileged mode'), while operating systems and related utilities may run in 'supervisor mode' (or 'privileged mode')" ("application program from FOLDOC", http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?application+program, accessed October 22, 2004). In contrast, a directory is "[a] node in a hierarchical file system, which contains zero or more other nodes – generally, files or other directories" ("directory from FOLDOC", http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?query=directory, accessed October 22, 2004).

In other words, a directory is part of the file system/operating system; an application program is not. The components of the market maker application of Meltzer, namely the communication agent and the various services are applications, and therefore are not the same things as a directory. The Examiner's argument analogizing an application to the directory of the claimed invention is therefore an inappropriate analogy. Meltzer fails to teach all of the features of the claimed invention, namely that the applications are interacting with a directory,

The Examiner argues that the market maker node of Meltzer is a distributed directory because the market maker node is a server (Office Action dated August 24, 2004, pages 8-9, ¶ 21), because it supports a repository structure. This argument has four flaws. First, it ignores the reality of the Meltzer system, in that the applications (that is, the services) are not interacting with the repository structure: they interact with the communication agent. Second, the Examiner is ignoring the significance of the term "distributed" in the claims. Third, Meltzer does not enable the concept of a "distributed directory". And fourth, if the fact that Meltzer teaches a server storing a repository structure enough to reject the claims in this application, then no patent could ever issue in the future on any file system.

The first argument presented above centers around the structure of the market maker node. As shown in FIG. 15 of Meltzer, the market maker node has a communication agent that acts as a go-between for the documents and the applications. In other words, the communication agent receives documents to be distributed to the applications, then transforms the documents and distributes them to various services. The operation of the communication agent is described in column 84, lines 16-63 of Meltzer.

The Examiner appears to be arguing that the communication agent of Meltzer is analogous to the distributed directory of the claims. But this is not accurate. The communication agent of Meltzer is an application itself, as it "is an Internet interface which includes appropriate protocol stacks supporting such protocols as HTTP, SMTP, FTP, or other protocols" (Meltzer, column 84, lines 25-28). This clearly describes an application, and not a distributed directory. Thus, Meltzer teaches a system and method for applications to interface with each other, but not a way for applications to interface with a distributed directory.

The second argument focuses on the fact that the claims describe the directory as "distributed". As described on pages 1-2 of the specification, Novell Directory Services (NDS) is an example of a directory service that "provides a logical tree-structure view of all resources on the network so that clients can access them without knowing where they are physically located." If there were only one computer involved, the client would know the physical location of the resource, but then again, the directory would not be distributed in that situation, either. Thus, a distributed directory involves resources that can be located on any computer in the network.

In contrast, the Examiner and Meltzer describe the market maker node as "a server". But if the market maker node is a server (singular), then it cannot be a distributed directory, which involves two or more computers. Although the Applicant believes that the user of "distributed directory" in the claims was sufficient to distinguish the invention over Meltzer, the Applicant has amended the claims to make explicit that the distributed directory involves two or more computers in a network. Since Meltzer teaches the market maker node as involving a single server, at best Meltzer teaches a directory, and not a distributed directory.

The third argument examines the fact that while the Examiner asserts Meltzer teaches a "distributed directory", the Examiner is forced to cite to Harrison to reject a claim specifically describing hardware supporting a distributed directory. In rejecting claim 18 and the claims that depend from claim 18, the Examiner has acknowledged that Meltzer does not teach the concept of the distributed directory being stored in the memories of different computers. But if Meltzer fails to teach or suggest this feature, then Meltzer does not provide an enabling description of the hardware necessary to implement a distributed directory. And if Meltzer fails to teach or suggest the necessary hardware as the Examiner admits, the Applicant believes Meltzer cannot teach or suggest a distributed directory as claimed in the remaining claims in the patent application, with or without a description of the hardware used to implement a distributed directory.

The fourth argument presented above is an argument relating to Meltzer as teaching a distributed directory is an argument relating to the fundamentals of the patent system. According to the Examiner, Meltzer teaches a server that has a repository structure, and that this is enough to teach the concept of a distributed directory. If the Examiner is correct, then no patent can ever issue in the future for a file system. After all, a file system is a way to store files so that applications can access the files. The specifics of how the file systems accomplish this object are irrelevant: what matters is their purpose. But according to the Examiner this purpose is described in Meltzer. Therefore, no future file system can ever be patented, as its objectives are already met. This attitude harkens back to Commissioner of Patents Charles H. Duell, who in 1899 remarked that "Everything that can be invented has been invented." The fact that a document (like Meltzer) teaches a concept similar to a claimed invention does not automatically mean the similar concept cannot be separately patented. Indeed, there are known examples of patents granted on new, unforeseeable uses for old objects. Clearly, that the distributed directory of the claimed invention is distinguishable from the repository structure of Meltzer (in its manner of interacting with applications, if in no other way) should distinguish the claimed invention sufficiently to make the claims in this patent application allowasble.

While it is true that some claims describe multiple applications, they are all interfacing with the directory, and not directly with each other. Even claims 29-30 and 41, which describe events occurring in both a directory and in two applications, still have the applications interfacing with the directory, and not with each other. Thus, the fact Meltzer teaches a system and method for applications to share data does not mean that Meltzer teaches or suggest applications interfacing with a directory.

Accordingly, Meltzer, Cronin, and Harrison fail to teach or suggest the features of the claimed invention, and claims 7, 9, 11, 15, 18-20, 22, and 26-41 are patentable under 35 U.S.C. § 103(a) over Meltzer in view of Cronin (and possibly Harrison), and claims 7, 9, 11, 15, 18-20, 22, 26-27, and 29-41 are allowable.

For the foregoing reasons, reconsideration and allowance of claims 7, 9, 11, 15, 18-20, 22, 26-27, and 29-41 of the application as amended is solicited. **Because this application is now past a second Request for Continued Examiner, the Examiner is requested to telephone the undersigned at (503) 222-3613 to schedule an interview.**

Respectfully submitted,
MARGER JOHNSON & McCOLLOM, P.C.

Ariel S. Rogson
Ariel S. Rogson
Reg. No. 43,054

MARGER JOHNSON & McCOLLOM, P.C.
1030 SW Morrison Street
Portland, OR 97205
503-222-3613
Customer No. 20575

I hereby certify that this correspondence is being transmitted to the U.S. Patent and Trademark Office via facsimile number 703/872.9306, on November 4, 2004

Jeanne Bower
Jeanne Bower